GridFTP based real-time data movement architecture for x-ray photon correlation spectroscopy at the Advanced Photon Source

S. Narayanan*, T.P. Madden, A.R. Sandy
Advanced Photon Source
Argonne National Laboratory
Argonne, IL, USA
*sureshn@aps.anl.gov

Raj Kettimuthu*, Michael Link
Mathematics and Computer Science
Argonne National Laboratory
Argonne, IL, USA
*kettimut@mcs.anl.gov

***Abstract*—X-ray photon correlation spectroscopy (XPCS) is a unique tool to study the dynamical properties in a wide range of materials over a wide spatial and temporal range. XPCS measures the correlated changes in the speckle pattern, produced when a coherent x-ray beam is scattered from a disordered sample, over a time series of area detector images. The technique rides on "Big Data" and relies heavily on high performance computing (HPC) techniques. In this paper, we propose a high-speed data movement architecture for moving data within the Advanced Photon Source (APS) as well as between APS and the users' institutions. We describe the challenges involved in the internal data movement and a GridFTP-based solution that enables more efficient usage of the APS beam time. The implementation of GridFTP plugin as part of the data acquisition system at the Advanced Photon Source for real time data transfer to the HPC system for data analysis is discussed.**

***Index Terms*—Big Data, GridFTP, High Performance Computing, X-ray Photon Correlation Spectroscopy, Synchrotron.**

I. Introduction

X-ray photon correlation spectroscopy (XPCS) is a powerful technique to probe the dynamics in materials over a wide range of length scales (micrometers-nanometers) and time scales (microseconds-hours) [1]. The dynamical phenomena that have been successfully studied using XPCS encompass the classical Brownian diffusion [2] in simple liquids to more complex hyper and sub-diffusive processes that have been observed in a host of complex fluids such as gels, emulsions, polymers [3]. One of the challenging areas of study that is of general interest and is being studied using XPCS is to measure the dynamical properties of concentrated eye-lens suspensions in order to help in understanding the effect of the changes in proteins on diseases such as presbyopia [4].

XPCS technique by functionality involves handling "Big Data" streaming at high data rates pushing the envelope of network bandwidth, disk writing and access speeds and high performance computing (HPC) for data analysis. The dynamical time scales that can be probed is limited at one end by how fast the detector can stream images and at the other end by the total number of images that are collected in a single data acquisition. The state-of-the-art CCD detector that is suitable for XPCS operates continuously at 60 fps (frames per second), streaming one million (1M) pixels producing 120 MB/sec of data [5]. New detectors that are suitable for XPCS and will be available in 2013 push the data rates significantly further by streaming 1M pixels at 200 fps and 22000 fps. This enables measuring the dynamics in samples at much shorter time scales, which are of relevance in understanding physiology in real conditions like cell membranes and eye lens [4].

The notion behind science in the fast track is to be able to acquire data and analyze in real time so that the experimenter is provided with a real time feedback on the physical parameters being measured. The experimenter benefits by being able to make quick changes to the experimental conditions for optimal applications. For data analysis, we apply HPC tools established at the APS using several compute nodes operating under a Lustre parallel file system. Often times, moving the data from the acquisition system to HPC cluster is the bottleneck.

In this paper, we propose a high-speed data movement architecture for moving data within the Advanced Photon Source (APS) as well as between APS and the users' institutions. We describe the development and deployment of a GridFTP-based solution for high throughput and high reliability real time data transfer from the data acquisition PCs to the HPC cluster. The paper describes the impact on data analysis by the progress made in the pipeline for data transfer from a traditional UNIX copy to an offline GridFTP to a real time data transfer by incorporating GridFTP with the data acquisition system.

The rest of the paper is organized as follows. In Section II, we provide background on the science done using XPCS. In Section III, we discuss the challenges in data movement faced at the XPCS beamline at APS. In Section IV, we provide background on GridFTP. We present a data movement architecture; describe the development and deployment of a GridFTP-based solution and its impact on beam time usage in Section V. In Section VI, we provide a brief description of the software package Experimental Physics and Industrial Control System (EPICS) that is used to control experiments and acquire data at XPCS. We present the design of GridFTP plugin for EPICS in Section VII. In Section VIII, we provide some results obtained using the plugin, future outlook and discuss how the solution deployed at XPCS can be applied at other beamlines and user facilities. We summarize in Section IX.

II. SCIENCE

XPCS probes the dynamics at nanometer length scales in an elegant way by looking at the ever-changing phase relationship between x-rays scattered from different points on the sample being measured. Random phase variations throughout the illuminated area on the sample give rise to constructive interference at some points in the far-field scattering pattern, and destructive interference at other points. This random interference is called a speckle pattern [6], and it is what is observed when a laser beam is reflected off of a rough surface. As the random phase arrangements change over time due to rearrangements within the sample owing to its dynamical nature, bright speckles become dark and dark speckles become bright. As a result, by studying the fluctuations in the speckle pattern with time, one can probe the time scale at which the medium undergoes changes. Further, by sorting speckle changes into different ranges of scattering angle (which has a direct correspondence on the length scales in the sample), one can directly study the different length scales over which the sample is changing. In practice, the sample being probed is illuminated with a coherent x-ray beam, and the coherent scattering pattern is collected on an area detector such as a charge-coupled device (CCD). By recording several thousands of image frames (totaling several gigabytes) over time, one can correlate the signal at each pixel over time, and do annular binning of the results to look at the length scale dependence. Thus in an x-ray photon correlation spectroscopy (XPCS) experiment, one measures the intensity-intensity autocorrelation function which can be related to the desired material properties like viscoelasticity [3].

The methodology of the data acquisition for XPCS and the computation that is involved is shown schematically in *Figure 1*. A time sequence of few thousand CCD images separated in time by a fixed time interval Δt are collected. A pixel-by-pixel correlation of the intensity at each pixel is performed with the corresponding pixel over the entire CCD frame (ca. 10^6 pixels) for different time delays (Δt and its integral multiples). The entire CCD frame is typically divided into annular bins whereby each annular bin corresponds to an average scattering angle (subtended by the group of pixels at the sample) or wave vector q.

$$g_2(\tau,q) = \frac{\langle I(t,q)I(t+\tau,q)\rangle}{\langle I(t,q)\rangle^2} = 1+\alpha\exp(-2\frac{\tau}{\tau_0(q)}), \qquad Eq.(1)$$

The intensity correlations across all the pixels that belong to the same annular bin or q is then averaged and normalized with the time average of the scattering intensity over the bin yielding $g_2(\tau,q)$ which contains dynamical information about the sample pertaining to that q as shown in Eq.(1). The scattering angle or wave vector q bears an inverse relationship to the length scale in the sample that is being probed. The g_2 typically exhibits a simple exponential decay with time in simple liquids exhibiting Brownian diffusion or more complex line shapes like a stretched or a compressed exponential in complex fluids like polymers [7] and gels [3]. Accordingly, the g_2 is modeled as shown in Eq.(1) yielding a characteristic relaxation time τ_0 at the specific q.

It can be clearly seen that performing the above computation on "Big Data" is a highly computationally intensive task. Since the computation of each pixel across time is independent of the other pixels, the process is fully parallelizable down to the level of a single pixel. XPCS analysis takes full advantage of this highly parallelizable computation process. A message passing interface (MPI) based correlation application has been developed which splits the data across multiple cores in a HPC environment to handle the above tasks [8]. A custom load balancing algorithm has been written which ensures that every core gets nearly equal number of pixels so as to minimize any delays in further processing steps.

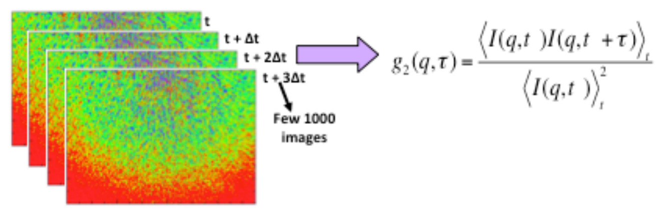

Figure 1: Schematic showing the methodology of XPCS data acquisition and computation of the intensity-intensity autocorrelation function.

A typical scientific example measuring the dynamics of a model system of polystyrene colloids of diameter 100 nm dispersed in a molecular liquid glycerol is presented here [9,10]. The data acquisition was carried out by operating the CCD at 60 fps collecting a total of 80000 frames spanning nearly five decades in time (ca.10^{-2}–10^{3} seconds) totaling 160 GB of raw data. It should be mentioned here that XPCS analysis takes advantage of the fact that the data in most cases is sparse and hence can be compressed reasonably well (ca.10-20x). The scattering pattern integrated over the entire CCD detector and over all the times (frames) is shown in Figure 2. The dark triangular object is the beam stop that blocks the very intense incident x-ray beam from hitting the detector which would otherwise damage the CCD. The scattering pattern is then measured on the CCD detector around the beam stop.

The measured scattering pattern shows a prominent strong ring of scattering that arises due to the interference between the colloidal spheres dispersed in glycerol and corresponds to an interparticle separation of 150 nm. The scattering pattern is azimuthally symmetric, with the center of the ring being the center of the incident x-ray beam corresponding to zero scattering angle or q=0. The autocorrelation functions g2 computed using the above procedure is shown in Figure 3 for three representative scattering angles or length scales. The relaxation time extracted from the fit based on Eq.(1) has a direct square dependence on the length scale probed which is attributed to the random walk like motion exhibited by the particles in a viscous liquid.

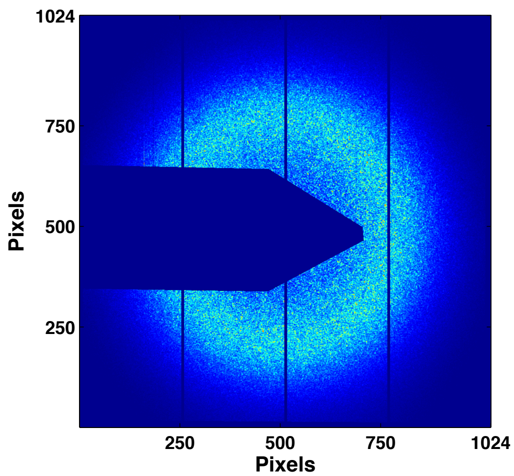

Figure 2: 2-D small angle x-ray scattering pattern measured from a colloidal suspension of 50 nm diameter polystyrene spheres dispersed in glycerol collected on a CCD detector. The dark region with a triangular tip is the shadow of the beam stop that is used to block the very intense incident x-ray beam from hitting the detector which would otherwise damage the CCD. The three vertical dark strips are the dead regions on the CCD associated with the different taps for faster readout.

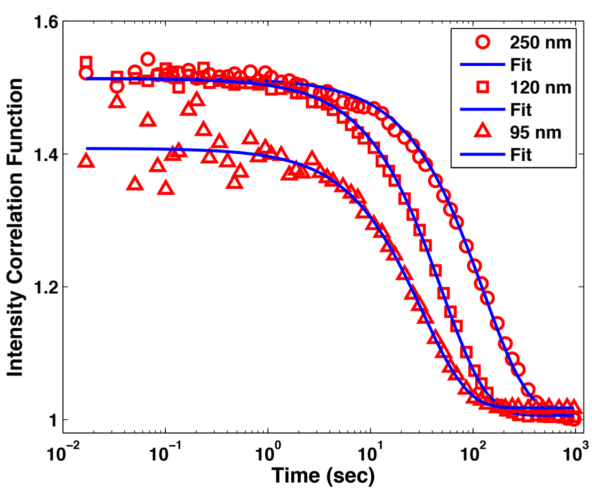

Figure 3: Intensity-Intensity autocorrelation function calculated using Eq.(1) from a series of 80,000 frames collected on a CCD detector running continuously at 60 frames per second. Measured g_2s at three different length scales on the sample are shown as symbols and the solid lines are fit to the data based on Eq.(1).

III. DATA MOVEMENT CHALLENGES

The main challenge in a typical XPCS experiment is to be able to measure and analyze the data using HPC so that the user gets rapid feedback on the results of the experiment. This requires that the data be transferred from the data acquisition PC to the HPC cluster at a high throughput. Initially, due to the lack of a high speed data transfer mechanism, a traditional mode of data transfer to the HPC was employed which involved the following steps. The raw data was first saved during the acquisition onto the local disk on the PC that controls the CCD detector. After the end of the acquisition of a few thousand frames totaling several gigabytes, a simple UNIX copy was performed onto a network file system (NFS) mount of the HPC Lustre file system. Due to the bottleneck of the disk access speeds from the local disk on the PC and the overheads associated with a simple copy, the data transfer rate was limited to ca. 10 MB/sec which resulted in a significant wastage of user beam time as the disk reading during the data transfer prevented any further data acquisition. This also prevented the measurement of samples that exhibit transient phenomena.

To overcome the above drawbacks and to be able to realize the ultimate goal of computing correlation functions in real time, we embarked upon GridFTP [11, 12] as a means for high throughput data transfer. The progress made and the scientific impact are discussed in the subsequent sections.

IV. GRIDFTP

The GridFTP file transfer protocol [11] extends the FTP file transfer protocol with features that permit high performance, secure and reliable data movement. It is based on the RFC 959 “File Transfer Protocol” [13], RFC 2228 “FTP Security Extensions” [14], and RFC 2389 “Feature Negotiation Mechanism for the File Transfer Protocol” [15]. Additionally, it defines a new data channel protocol called extended block mode [11]. The GridFTP protocol has been standardized through the Open Grid Forum. The standardization has led to the development of multiple interoperable implementations. The Globus implementation of GridFTP [12] is the most widely used.

The following is a summary of the key features of GridFTP:

• GridFTP allows third party control of data transfer whereby a user can initiate, monitor and control data transfers between two remote machines which are the source and destination for the data transfer.

• GridFTP supports Generic Security Services (GSS)-API authentication of the control channel [14] and data channel (GridFTP extensions) [11], and supports user controlled levels of data integrity and/or confidentiality. Data channel authentication is of particular importance in third party transfers since the IP address of the host connecting for the data channel (i.e., the other storage system) will be different than that of the host connected on the control channel (i.e., the client), and there must be some way to verify that it is the intended party.

• On wide-area links, using multiple TCP streams in parallel between a single source and destination can improve the aggregate bandwidth relative to that achieved by a single stream [16, 17]. GridFTP supports such parallelism via FTP command extensions and data channel extensions.

• GridFTP supports striped data movement operations, in which a set of computers is used in a coordinated fashion to move data from one parallel file system to another. Note that striping and parallelism may be used in tandem, i.e., you may have multiple TCP streams open between each of the multiple servers participating in a striped transfer.

• Some applications can benefit from transferring portions of files rather than complete files: for example, analyses that require access to subsets of massive files. FTP allows transfer of the remainder of a file starting at a specified offset. GridFTP supports requests for arbitrary file regions.

• Reliable transfer is important for many applications that manage data. Fault recovery methods are needed to handle failures such as transient network and server outages. The FTP standard includes basic features for restarting failed transfers that are not widely implemented. GridFTP exploits these features and extends them to cover its new data channel protocol.

V. GRIDFTP BASED DATA TRANSFER

The automated data movement architecture for XPCS is based on GridFTP, Globus Online [18, 19] and MyProxy [20]. APS beamline resources and HPC resources are protected by strong firewall rules that prevent access to the outside world for security reasons – whereby both inbound and outbound connections are not allowed. Data movement for beamline experiments can be classified into two categories: internal and external. Internal data movement corresponds to the data movement from the beamline data acquisition machine to the HPC cluster for online data analysis. External data movement corresponds to the transfer of data from APS to user’s home institution for offline data analysis.

We have developed a GridFTP based solution for both internal and external data movement. Since the internal data movement happens in a strict firewall constrained environment, traditional GridFTP client based “globus-url-copy” is used for data movement. Traditionally, the external data movement amounting to several terabytes per user per experiment was done by shipping USB hard disks. In order to facilitate the use of network for external data movement, we have proposed to run a GridFTP server in a demilitarized zone (DMZ) outside the APS Tier-2 firewall (*Figure 4*). The DMZ in this case is still within the Argonne Tier-1 firewall with appropriate ports opened for GridFTP data movement. APS Lustre parallel file system is accessible from the GridFTP server on DMZ with read-only permissions. Since the GridFTP server on DMZ is accessible from the outside world, Globus Online (hosted GridFTP client) is recommended for the external data movement. We are still working through security aspects (authentication and authorization) for external data movement pertaining to the unique credentials for each XPCS user. So, the focus of this paper is on the internal data movement. In the following sub-sections, the GridFTP-based solution deployed for XPCS data movement within APS and the methodology of the transition from a post acquisition to a real time data transfer, both based on GridFTP is discussed.

A. GridFTP for Windows

Globus implementation provides GridFTP server, client and development libraries. Even though the client libraries were available for both Windows and UNIX-based platforms, the server was originally available only for UNIX-based platforms. XPCS use Microsoft windows computers for data acquisition and the transfer of data from the acquisition machine to the HPC cluster must be controlled from a remote machine.

1) Cygwin-based GridFTP

The data generated in XPCS experiments must be moved to a HPC cluster after every data acquisition. Typically, this data transfer is done via network file system (NFS) mounting, which is slow. During this time the acquisition machine is either idle or underperforming due to ongoing data transfer. We ported the Globus GridFTP server to Cygwin, a Linux-like environment for Windows and deployed it on the beamline data acquisition machines for X-ray photon correlation spectroscopy. Using a single-user GridFTP server for Cygwin, we have demonstrated that GridFTP can improve the performance of this data movement from acquisition machine to HPC cluster by up to a factor of eight. But the Cygwin approach has a number of limitations as discussed below.

2) Native implementation of GridFTP for Windows

A GridFTP server running natively on Windows would significantly reduce the dead time during experimental runs for all these user facilities. Scientists can start processing data sooner, leading to better quality science due to the ability to make decisions on results of recently run samples.

Cygwin is not a way to run native Linux apps on Windows. You have to rebuild your application from source if you want it to run on Windows. A Cygwin-based GridFTP is slower than a native Windows implementation with file accesses in particular taking much longer. It also creates issues in dealing with management of the Cygwin framework using dynamically linked libraries (DLL) when other Cygwin apps are running on the same machine, as there are issues with having multiple Cygwin DLL’s on a machine. It also does not easily solve the identity/privilege problem, as Cygwin requires an additional plugin to be installed into the security subsystem of the machine in order for functions like setuid() to work. Additionally, there are other quirks such as not getting the correct file stat info when accessing files via their windows path as opposed to a POSIX path. For a fully functioning windows native server, we would need to implement a solution for forking new connections (Windows does not support traditional fork() calls) and process identity/privilege settings (since there is no direct replacement for setuid() in Windows) and file system access control. There would also need to be a slight redesign in the way we handle file metadata containing file ownership and permission information throughout the server.

Recently (in July 2012), we have developed a single user native GridFTP server for Windows that would solve the problems of the single user Cygwin server. Further, we are working on developing a native multi-user GridFTP server for Windows.

B. Post Acquisition GridFTP based Data Transfer

GridFTP server for Windows is deployed on the data acquisition system and GridFTP server for Linux is deployed on the HPC cluster. The command-line GridFTP client globus-url-copy is used to move data between the two servers. The application of GridFTP made a huge improvement in the data transfer rate. By setting up anonymous GridFTP servers on both the detector PC as well as on the HPC system, the data acquisition automatically initiates a GridFTP transfer right after the acquisition is done which occurs at an average rate of ca. 80 MB/sec over a 1 Gbit Ethernet link. This resulted in ca. 8x reduction in the data transfer time which directly translated to a proportionally more efficient usage of the APS beam time and more productivity.

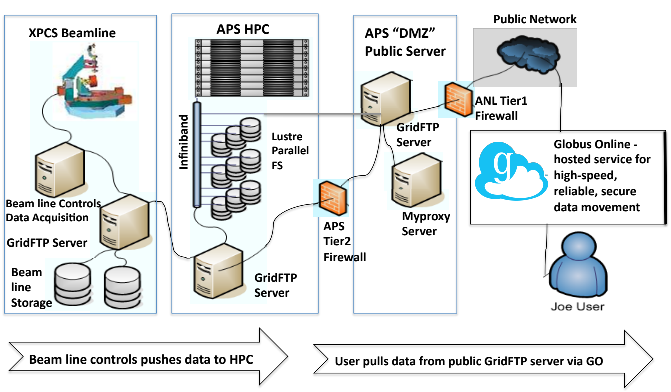

Figure 4: Automated data movement architecture for XPCS at the APS.

C. Real time GridFTP based data transfer

The major drawback of the above analysis method is that the user has to make a guess on the number of frames that need to be acquired for a decent signal to noise ratio (SNR) of the correlation functions. The additional complication comes from the fact that the samples undergo radiation damage over time with exposure to the x-rays. Due to the above issues, the user typically acquires an arbitrary number of CCD images and then processes the data to examine the SNR based on which the user has to make a decision on the subsequent acquisition steps. The desired approach and the optimum use of the beam time is to stream images from the CCD detector to a HPC cluster in real time such that the cluster can start processing the data in real time. This will enable the user to visualize the correlation functions in real time as more images are acquired and the user can make a judicious decision about the optimal SNR so that the data acquisition can be aborted. In order to facilitate this, we embarked upon integrating GridFTP as part of our data acquisition system as a high throughput and reliable data transfer mechanism. The implementation and the performance metrics are discussed in the following sections.

VI. EPICS AREA DETECTOR

A software package called *Experimental Physics and Industrial Control System* (EPICS) is used at the APS and many user facilities to control experiments and acquire data. At the XPCS beamline at Sector-8, EPICS Area Detector (EPICS-AD), a specific EPICS-based application, is used to control area detectors including setting up detector parameters, triggering exposures and collecting image data. The EPICS-AD application for the DALSA CCD runs on a Windows PC and controls a DALSA Camera-Link based frame grabber connected to the CCD detector. The EPICS-AD software triggers the camera and acquires images into system memory via the frame grabber. EPICS can then save images to files and allow for image display through the use of *plugins*. A plugin is a software extension to an EPICS-AD application to add specific functionality such as saving data of some file format, performing real-time statistical calculations on the images, or posting the image data over the network port for real-time display by any remote client like MATLAB, Python, IDL.

The original software design was for EPICS-AD to stream frames from the detector to the disk on the local PC in a custom binary format called *IMM* using a plugin such that multiple images are appended to the same file along with the metadata. Once the data are stored to the disk, a GridFTP client transfers the file from the disk to a remote anonymous GridFTP server on a *High Performance Computing* (HPC) cluster (Figure 5). The limitations of this mode are two fold: (i) the speed of the GridFTP transfer is limited by the local disk, and (ii) the file transfer cannot commence until the acquisition of an entire dataset consisting of 1000's of frames (several GB of data) is completed.

To overcome this bottle neck, we designed a GridFTP plugin to run as part of the EPICS-AD application that streams frames directly from the system memory to the remote anonymous GridFTP server on the HPC cluster. Because it is desired to store the data in IMM format on the remote server, the GridFTP plugin gets its input from the IMM file-saving plugin, which structures the data and the metadata in IMM format. The IMM plugin can optionally save the data to the local disk simultaneously as the GridFTP plugin sends IMM-formatted data to the remote GridFTP server. In this mode, the bottle necks associated with the previous mode are overcome (Figure 6).

VII. DESIGN OF GRIDFTP PLUGIN

EPICS-AD is written in C and C++ and runs on several operating systems. Windows OS is used at Sector-8 because the frame grabber API is supported only on Windows. Further, because the GridFTP C++ API is not compatible with Windows, the GridFTP Java API was used. First, a GridFTP client was written in Java to transfer images from the Java memory space to the remote GridFTP server. Further, the Java client can create directories on the remote server and check for the existence of the directory. The next step was to connect the Java GridFTP client to the C++ EPICS-AD software as an EPICS plugin. This was accomplished by designing a very general-purpose plugin, called CJAreaDetector written in C++ that uses the Java Native Interface (JNI) to start a Java Virtual Machine as part of the EPICS process, load a Java class, and run the class. A Java class, JAreaDetector, was created to act as an interface between the GridFTP client and EPICS. CJAreaDetector always loads a class derived from JAreaDetector and uses JAreaDetector methods to transfer images between the Java and C++ memory spaces. The GridFTP client is written as a subclass of JAreaDetector. In general, any functionality can be written in Java, and interfaced with EPICS using the JArea Detector class as an interface. To prevent Java code from blocking the EPICS C++ code, the JAreaDetecor runs on its own thread.

A. Connection of Plugins

EPICS-AD plugins can be reconfigured at run-time. In this application, the preferred configuration is to send raw images from the detector to the IMM plugin to convert the data to IMM format and add metadata information. The data is then piped to the GridFTP plugin for storage on the remote server. However, in general, the developed GridFTP plugin can be configured to accept data directly from the EPICS-AD detector driver that streams raw data or from any other file plugin like TIFF, HDF5, so long as the other file plugins are configured to stream data.

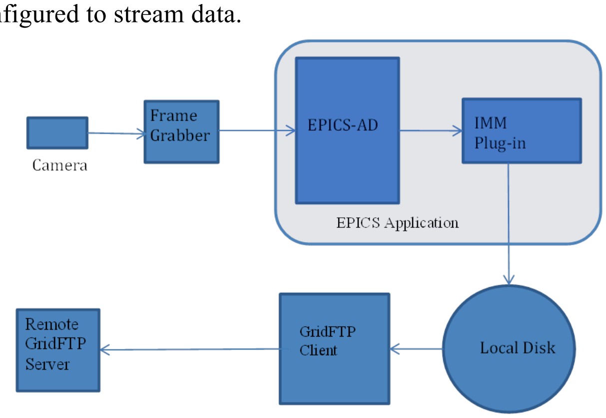

Figure 5: EPICS-AD collects data from camera and stores to the disk. A separate application reads image data from the disk and transfers to the remote GridFTP server. In this scenario, a write to disk and a read from the disk is required for each data transfer.

Each plugin has its own set of control parameters such as the directory name, file name and compressed/uncompressed format. The GridFTP plugin has configurable parameters for the remote server name and TCP port, remote directory name and the remote file name. Additionally, the plugin also performs the task of checking the existence of remote directory location and creation. To simplify the user interface, the GridFTP plugin can inherit its parameters from the IMM plugin. In this way, the user sets the filename on the IMM plugin and it is saved with the same name on the remote GridFTP server. The user need only set the server name, remote directory name, and a flag to enable data transfer of all IMM data. A schematic of the GridFTP plugin GUI integrated with the data acquisition system is shown in Figure 7.

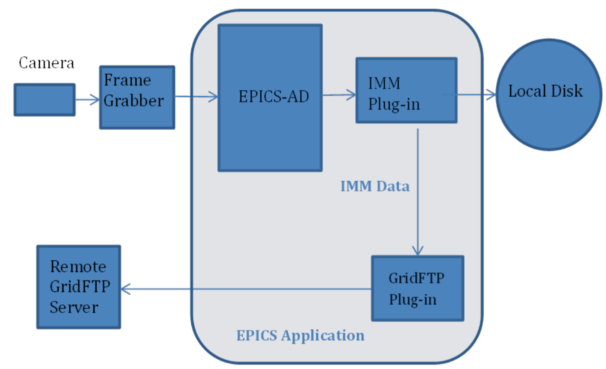

Figure 6: EPICS-AD application gathers the data from the detector. The IMM plugin formats the images into IMM format and appends metadata information. Further, the data is sent simultaneously to the local disk and to the GridFTP plugin, that sendsIMM-formatted data to the remote GridFTP server. In this scenario, a write to the local disk is not required for the GridFTP transfer, as data is sent directly from the memory.

Figure 7: Graphical user interface of the GridFTP plugin integrated with the EPICS-AD acquisition system.

VIII. Results and Future Outlook

The pile of "Big Data" under which the subtleties of dynamic phenomena under investigation are hidden is brought into light using a combination of high performance computing tools combined with GridFTP based high throughput data transfer mechanism. The pipeline of the data flow from the acquisition to the HPC system was improved in the following steps: (i) post acquisition traditional UNIX copy, (ii) post acquisition GridFTP based data transfer, and (iii) real time data transfer using GridFTP plugin incorporated within the data acquisition system. The transition from (i) to (ii) resulted in an improvement in the data acquisition efficiency by 8x directly translating to an efficient usage of the APS beam time which is a significant resource.

The real time data transfer using GridFTP plugin has been successfully tested at frame rates up to 60 fps whereby there is no noticeable latency or any dropped frames between the data saved to the local disk on the acquisition PC and the data transferred to the HPC cluster. Even though the current network bandwidth between the XPCS beamline and the HPC is 1 Gbit/s which is close to the data rate at 60 fps, the fact that the data can be made sparse before being written to the disk circumvents the problem to some extent. The next generation fast CCD detector that will be commissioned in 2013 will have a data rate that is ca.10x due to a combined effect of higher frame rate and higher quantum efficiency. We plan to upgrade the link between XPCS beamline and the HPC to 10 Gbit/s. Once the hardware limit is increased, we plan to re-evaluate the plugin and make necessary improvements to utilize the 10 Gbit/s link efficiently.

As the GridFTP server for windows is implemented as open source and available for production use, any facility with windows machine for data acquisition can make use of it. The real-time high throughput and high reliability data transfer model is based on combining GridFTP as an integral part of the open source EPICS based data acquisition system for the XPCS scientific program at the APS. This successful solution can be easily adopted by other "Big Data" based scientific programs at the APS other user facilities around the world that use EPICS for data acquisition.

IX. Summary

We have presented details of X-ray photon correlation spectroscopy (XPCS), a unique tool used to understand the dynamical properties in a wide range of materials, its scientific importance and the data movement challenges. We have also presented a GridFTP-based data movement solution for moving data from data acquisition system to the HPC cluster that improves the data movement rate by a factor of 8 compared to using a traditional copy by mounting the remote file system. Finally, we described the design of a GridFTP plugin for the experiment control and data acquisition system EPCIS to yield correlation functions in real-time for XPCS at APS and how the solutions developed here will benefit other beam lines at APS and other national user facilities.

Acknowledgment

The authors thank Brian Tieman for the initial implementation of GridFTP data transfer at the APS and Roger Sersted for the installation and reliable maintenance of GridFTP for regular operations. Use of the Advanced Photon Source, an Office of Science User Facility operated for the US Department of Energy (DOE) Office of Science by Argonne National Laboratory, was supported by the US DOE under Contract No. DE-AC02-06CH11357.

References

[1] R. G. Larson, The structure and rheology of complex fluids, Oxford University Press, (1999).

[2] A. Einstein, *Investigations of the Theory of Brownian Movement* (Dover, 1956).

[3] Anders Madsen, Robert L Leheny, Hongyu Guo, Michael Sprung and Orsolya Czakkel1, Beyond simple exponential correlation functions and equilibrium dynamics in x-ray photon correlation spectroscopy, New J. Phys. 12, 055001 (2010).

[4] Jirka Peschek, Nathalie Braun, Titus M. Franzmann, Yannis Georgalis, Martin Haslbeck, Sevil Weinkauf, and Johannes Buchner, The eye lens chaperone α-crystallin forms defined globular assemblies, PNAS 106, 13272 (2009).

[5] P. Falus, M. A. Borthwick, and S. G. J. Mochrie, Fast CCD camera for x-ray photon correlation spectroscopy and time-resolved x-ray scattering and imaging, Rev. Sci. Instrum. 75, 4383 (2004).

[6] M. Sutton, S. G. J. Mochrie, T. Greytak, S. E. Nagler, L. E. Berman, G. A. Held, G. B. Stephenson, Observation of speckle by diffraction with coherent X-rays, *Nature* 352, 608 (1991).

[7] Hongyu Guo, Gilles Bourret, Muriel K. Corbierre, Simona Rucareanu, R. Bruce Lennox, Khalid Laaziri, Luc Piche, Mark Sutton, James L. Harden, and Robert L. Leheny, Nanoparticle Motion within Glassy Polymer Melts, Phys. Rev. Lett. 102, 075702 (2009).

[8] Brian Tieman, Suresh Narayanan, Alec Sandy, Marcin Sikorski, MPICorrelator: A parallel code for performing time correlations, Nuclear Instruments and Methods in Physics Research A 649, 240 (2011).

[9] L. B. Lurio, D. Lumma, A. R. Sandy, M. A. Borthwick, P. Falus, S. G. J. Mochrie, J. F. Pelletier, M. Sutton, Lynne Regan, A. Malik, and G. B. Stephenson, Absence of Scaling for the Intermediate Scattering Function of a Hard-Sphere Suspension: Static and Dynamic X-Ray Scattering from Concentrated Polystyrene Latex Spheres, Phys. Rev. Lett. 84, 785 (2000).

[10] M. Sikorski, A. R. Sandy, and S. Narayanan, Depletion-Induced Structure and Dynamics in Bimodal Colloidal Suspensions, Phys. Rev. Lett. 106, 188301 (2011).

[11] Allcock, W. GridFTP: Protocol Extensions to FTP for the Grid. GFD-R-P.020, Global Grid Forum, 2003.

[12] Allcock, B., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I. and Foster, I., The Globus Striped GridFTP Framework and Server. SC'2005, 2005.

[13] Postel, J. and Reynolds, J. File Transfer Protocol. Internet Engineering Task Force, RFC 959, 1985.

[14] Horowitz, M. and Lunt, S. FTP Security Extensions. IETF RFC 2228, 1997

[15] Hethmon, P. and Elz, R. Feature negotiation mechanism for the File Transfer Protocol. Internet Engineering Task Force RFC 2389, August 1998.

[16] Hacker, T., Athey, B. and Noble, B., The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network. 16th International Parallel and Distributed Processing Symposium, 2002

[17] Qiu, L., Zhang, Y. and Keshav, S., On Individual and Aggregate TCP Performance. 7th International Conference on Network Protocols, 1999.

[18] Allen, B., Bresnahan, J., Childers, L., Foster, I., Kandaswamy, G., Kettimuthu, R., Kordas, J., Link, M., Martin, S., Pickett, K. and Tuecke, S. Software as a Service for Data Scientists. *Communications of the ACM*, 55(2):81-88, 2012.

[19] Foster, I. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing*(May/June):70-73, 2011.

[20] [20] Novotny, J. et al An Online Credential Repository for the Grid: MyProxy. 10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, 2001.